PATENT

AF?

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant:	Finlay et al.	Examiner:	Anh Ly
Serial No.	09/628,599	Group Art Unit:	2172
Filed:	July 28, 2000	Docket No.	CA990018US1
Title:	DIRECT CALL THREADED CODE		

CERTIFICATE UNDER 37 CFR 1.8

I hereby certify that this correspondence and identified enclosures are being deposited with the United States Postal Service, first class mail, postage prepaid, under 37 C.F.R. § 1.8 on the date indicated below and is addressed to the Mail Stop: Appeal Brief-Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, on January 3, 2007.

Sandra Parker

APPEAL BRIEF

Mail Stop: Appeal Brief-Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Brief is submitted pursuant to the Notice of Appeal filed on October 31, 2006 and is submitted timely because January 1, 2007 was a national holiday and January 2, 2007 was declared a Federal Holiday by the USPTO, which closed the USPTO on that day.

The Commissioner is hereby authorized to charge payment of the $170, which is the difference of the new appeal brief fee of $500 and $330 appeal brief fee paid on September 8, 2004 before Examiner reopened the prosecution, and any additional fees required for the above-identified application or credit any overpayment to Deposit Account No. 09-0460.

01/08/2007 CCHAU1 00000034 090460 09628599
01 FC:1402 500.00 DA

1. Real Party in Interest

The real party in interest in this appeal is International Business Machines Corporation of Armonk, New York and the assignee.

2. Related Appeals and Interferences

There are no other appeals or interferences known to the appellant, the appellant's legal representative or the assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

3. Status of Claims

Claims 1-22 remain pending in the application and are the subject of this appeal. Claims 1, 10 and 19 were previously amended on June 4, 2003 before an RCE was filed. A copy of all pending claims, including the claims on appeal, are set forth in an attached Appendix.

4. Status of Amendments

Response to Final Office Action containing a request for reconsideration and submitted on June 2, 2004 has been entered, as indicated in the Advisory Action of July 15, 2004.

5

However, the Amendment After Final dated October 31, 2006, containing amendments requested by the Primary Examiner Corrielus in July and August of 2006, was not entered by Examiner Ly, after Examiner Corrielus was suddenly reassigned, according to the Advisory Action dated November 8, 2006.

5. Summary of the Claimed Subject Matter

All independent claims 1, 10 and 19 of the present invention are specifically directed to show an improvement of a standard database management system which includes an implementation of a direct call mechanism replacing the lookup function of a run-time interpreter and a method for pre-processing an already created access plan to provide a direct call mechanism in such a system. This improvement is described in Title, Figs. 1-4 and Specification on p. 2, li. 16-31; p. 3, li. 2-10, p. 6, li. 14-30, pages 7-9, and is used to provide faster access which is cost-effective.

As shown throughout the Specification, the present invention is directed to RDBMSes with Interpreters and, specifically on pp. 3-4, it is shown that the invention relates to a system, program storage device and method for pre-processing an access plan generated for a query in a relational database management system to include a direct call mechanism replacing a lookup function of a run-time interpreter. The access plan includes a plurality of operation codes (op. codes), each of the operation codes being associated with one or more executable functions for performing the query.

Specifically, the independent claim 1 states:

1. A method for pre-processing an access plan generated for a query in a relational database management system to include a direct call mechanism replacing a lookup function of a run-time interpreter, said access plan including a plurality of operation codes, each of said operation codes being associated with one or more executable functions for performing the query, said method comprising the steps of:

(a) determining from the access plan an executable function associated with a first operation code; and

(b) augmenting said first operation code in the access plan with a pointer to said executable function to provide a direct call mechanism replacing a lookup function of a run-time interpreter.

As can be seen, claimed method is for pre-processing of an existing access plan which has op. codes and is to be interpreted by a Software Interpreter (not a Compiler). Step (a) determines, for an op. code from the existing access plan, an associated executable function which should be

interpreted if there is no present invention. Step (b) augments the op. code, inside the existing access plan, as shown in Fig. 3 in block 208 and Specification on p. 3, li. 4-10, p. 7, li. 3-22, p. 8, li. 2-16, and disallows interpretation of the executable function by the Interpreter because it replaces the op. code with a direct pointer to the function, located outside the access plan, which implements the operation indicated by the op. code, thus removing the interpretive step. This means that later on, when the access plan is being executed at run-time, instead of interpreting the op. code, the pointer is used to call the replacement code, thus a term "direct call", which is described in much more detail in the Specification on pp. 7-9.

Thus, the claims are directed to preprocessing, which happens prior to execution, which will allow a run-time improvement of execution of an existing and previously optimized access plan whose steps have already been determined and are not changed by the present invention but are substituted with pointers (Spec. p. 7, li. 5-10) in order to increase the run-time speed. As can be seen on p. 7, li. 25, the pre-processing is performed by the Access Plan Manager prior to storing the improved access path plan into a memory cache and before the execution. As can be seen on p.8, li. 14-16, this processed code section has pointers and is ready for execution, as described on p. 8, li. 17-29.

The pre-pass mechanism of to the present invention replaces the repeated looking up of the function to call to process the OPCODE and the function's operands, and any decisions that need to be made repeatedly (i.e. static decisions), during the interpreter phase of execution. The pre-pass mechanism comprises a pre-processing function which replaces or augments the OPCODE, and any static decisions, with a pointer to the function to call to perform the operation specified by the OPCODE, or a pointer to an intermediate function with an auxiliary data structure, or a pointer to an auxiliary data structure, wherein the auxiliary data structure includes a pointer to the function to call to perform the operation specified by the OPCODE, as shown in Specification on p. 7, li. 3-22 and p. 8, li. 1-16. Advantageously, the pointers are called without additional lookup. The intermediate function to call to perform the function specified by the OPCODE may include processing operations and static decision making.

Specifically, p. 2., li. 1-15 of Specification describe a creation of an access plan with op. codes, p. 2, li. 16-31 describe a prior art problem because an interpreter has to translate and run each op. code one at the time, and during that step interpretation it has to examine the op. code, look up an

associated function which is called to process that op. code, make a decision in order to interpret and process the op. code and then run the op. code. On p.3, in the Summary section quoted above, it is shown that the present invention includes a pre-processing step which replaces each access plan op. code with a pointer to its function to provide a direct call, to avoid the lookup, wherein the pointer is named "a direct pointer". This is described in detail in pp. 7-9 of the Specification.

6. Grounds of Rejection

The claims are on Appeal for the following grounds of rejection:

(A) Claims 1-8, 10-17 and 19-22 stand rejected under 35 U.S.C. 102(e) as being anticipated by US Patent No. 5,875,334 issued to Chow et al. ("Chow").

(B) Claims 9 and 18 stand rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No.: 5,875,334 issued to Chow et al. (hereinafter Chow) in view of US Patent No.: 6,077,312 issued to Bates et al. ("Bates").

7. Arguments

This Appeal Brief is directed to Office Action dated July 31, 2006 which takes rejections verbatim from the Office Action dated February 7, 2006, both sent after the Applicant's Appeal Brief of September 8, 2004, when the Examiner reopened prosecution with all new references.

(A) Sec. 102(e) Rejection of Claims 1-8, 10-17 and 19-22

Sec. 102(e) Rejection of Independent Claims 1, 10 and 19

i) Examiner has Misinterpreted the Limitations of the Claimed Invention and Prior Art

Applicant regrets that all his attorney's attempts to obtain an examiner's interview with Examiner Ly and explain misinterpretation of certain computer science terms in office actions were unnecessarily turned down, despite her numerous attempts to talk to the Examiner and his supervisor before and after filing the first appeal in 2004, thus prolonging prosecution and causing unnecessary costs. Each Response to an Office Action clearly explained each misinterpretation but to no avail, because they were again repeated verbatim. Filing an appeal in 2004 was supposed to stop this practice but to no avail because the Examiner again reopened examination instead allowing the Board to decide on the merits.

As repeated by Applicant numerous times during phone calls and in e-mails, the invention has not been properly understood. Claims and specification were amended for clarification, only because they were requested in the Continuation of Substance of Interview by Primary Examiner Corrielus on 7/31/2006 in order to allow this application, but this amendment was not entered by the Examiner and the notice of allowance was not issued. Instead of allowing a telephonic conference which would clarify the claimed invention and prior art, the examiners issued a Final Office Action in which the Response to Argument section, p. 2-3, is a repetition of the rejections and which further failed to clarify the rejections and prior art. It is vague and broad and failed to directly respond to Applicant's arguments from the Response to the first office action after appeal, dated 5/8/2006, and to clearly point out where in Chow reference are found which elements used for 102 and 103 rejections. Moreover, it repeated misinterpretations and did not interpret the claims in light of the specification and drawings. Thus, the Final Office Action, as a whole, failed to address Applicant's numerous arguments about apparent misunderstanding of the claimed present invention and prior art. It also failed to clarify ambiguities regarding sited elements of prior art, such as which claimed element of the present invention is allegedly taught by which prior art element, as pointed out in the Response.

With respect to claims 1, 10 and 19, Office Actions stated: "Chow teaches a method for pre-processing an access plan generated for a query in a relational database management system to include said access plan including a plurality of operation codes, each of said operation codes being associated with one or more executable functions for performing the query (figs. 1-3, Pre-processing a SQL query statement is processed through a query compiler for generating a access executable plan for a database system such as IBM's DB2; this process requiring a run-time interpreter for looking up local variable; also this executable plan includes a plurality of SOD Query control statements, which are parsing into some operation codes into query graph model (QGM) representation of the statements, from which they are then processed to optimized QGM, an access execution plan such as cost-based optimized access plan and they are to be produced operation code or opcode by code generation module (item 118): abstract, col. 1, lines 15-35, col. 6, lines 25-67 and col. 8, lines 8-58; also col. 10, lines 1-40); determining from the access plan an executable function associated with a first operation code (fig. 1, parser, the optimized QGM, and threaded code generation are used to produce operation code of SQL query statements to get the optimized access plan or access execution plan : figs. 1 and 2, item 115 & 118, col. 6, lines 25-67 and col. 8, lines 8-58, col. 14, lines 40-67 and col. 15, lines 1-67 and col. 16, lines 1-67); and augmenting said first operation code in the access plan with a pointer to said executable function to provide a direct call mechanism replacing lookup function of a runtime interpreter (figs. 1 & 2, the extracted SQL statement or control statements is used to produce opcode based on parser, QGM and code generation and replacing this opcode (fig.2 and col. 15, lines 45-67 and col. 17, lines 25-67); the executable function in the run-time interpreter with a compiler generated local variables storing in symbol table for looking up with referencing local variables or pointers or indexes: fig. 4, Col. 24, lines 36-67 and col. 25, lines 1-8; also, col. 15, lines 1-67, col. 17, lines 1-28, Col. 20, lines 1-45 and col. 21, lines 5-55)."

The Applicant respectfully objects to the misinterpretation that Chow teaches "a method for pre-processing an access plan generated for a query in a relational database management system to include said access plan including a plurality of operation codes, each of said operation codes being associated with one or more executable functions for performing the query" because Chow does not have the quoted language. It is the language taken verbatim from the preamble of claim 1 of the present invention.

The Applicant respectfully objected to numerous deliberate misinterpretations found in this rejection and throughout the Office Actions for all rejected claims because the cited reference does not have the quoted language cited before the brackets and cannot perform the claimed steps. It is the language taken verbatim from the claims of the present invention, with reference page and column numbers added in brackets. However, Examiner continued with the practice. This way of writing is impermissible. PTO rules require use of English language and, thus, the proper English grammar, which was not followed in the Office Actions. Such use is a material misinterpretation of the reference's teachings. Moreover, the Chow reference does not contain teaching of this language. Therefore, Chow references was misinterpreted and deemed to teach what was not shown in the reference. Thus, that Final Office Action statement and rejection is incorrect.

Further, all Office Actions use the practice, subject to abuse, which disregards the law because in them some parts of claim 1 and even parts of claimed steps (a) and (b) were considered in isolation. For example, this rejection failed to include the language from lines 2-3 of claim 1 of the present invention, namely, "direct call replacing a lookup function of the run time interpreter". Examiner should not disregard express limitations, literally recited in the claim and disclosed in specification and has to specifically point out to it in the reference. In re Gulack, 703 F.2d 1381, 1384 (Fed. Cir. 1983) the court stated that the claim must be read as a whole and software limitations cannot be dissected from the prior art to support a rejection under 35 USC 103.

The rule is even more stringent for a 102(e) rejection, used for this application, because to establish prima facie case of anticipation of a claimed invention the standard for determining novelty under 35 USC 102 was been set forth by the courts *In re Bartleb*, 300 F.2d 942, 133 USPQ 204 (CCPA 1962), and *Hupp v. Siroflex of America Inc.*, 122 F.3d 1456, 43 USPQ2d 1887 (Fed. Cir. 1997), which states that the reference must be identical in all material aspects. However, this rule was not followed in the Office Actions.

The claim may not be dissected into arbitrarily selected discrete elements to be analyzed in isolation but must be considered as a whole. This law was not followed in Office Actions. Claim 1 was dissected in arbitrary pieces, alleging that the language was found in the prior art, although it was not. Neither preamble nor steps (a) and (b) limitations of claim 1 were reviewed in

entirety. They were dissected, some parts were removed, and then each remaining part was reviewed in isolation. Thus, claim 1 was not reviewed properly as a whole and words were taken selectively, picked and deleted without clear showing of such actions, which is an intentional misinterpretation of the claims. Furthermore, claim elements were placed next to Chow reference columns although they do not exist in the reference. Therefore, the Office Action statement and rejection of claims 1, 10 and 19 is incorrect.

All independent claims 1, 10 and 19 of the present invention are specifically directed to show an improvement of a standard database management system which includes implementation of a direct call mechanism replacing the lookup function of a run-time interpreter and a method for pre-processing an already created access plan to provide a direct call mechanism in such a system. They recite novel structure and thus distinguish over the cited prior art, under 35 U.S.C. 102(e) and 103(a). This is described in Title, Figs. 1-4 and Specification on p. 2, li. 16-31; p. 3, li. 2-10, p. 6, li. 14-30, pages 7-9, and is used to provide faster access which is cost-effective.

Specifically, the independent claim 1 states:

1. A method for pre-processing an access plan generated for a query in a relational database management system to include a direct call mechanism replacing a lookup function of a run-time interpreter, said access plan including a plurality of operation codes, each of said operation codes being associated with one or more executable functions for performing the query, said method comprising the steps of:

(a) determining from the access plan an executable function associated with a first operation code; and

(b) augmenting said first operation code in the access plan with a pointer to said executable function to provide a direct call mechanism replacing a lookup function of a run-time interpreter.

As can be seen, the claimed method is for pre-processing of an existing access plan which has op. codes and is to be interpreted by a Software Interpreter (not a Compiler). Step (a) determines, for an op. code from the existing access plan, an associated executable function which should be interpreted if there is no present invention. Step (b) augments the op. code, inside the existing access plan, and disallows interpretation of the executable function by the Interpreter because it replaces the op. code with a direct pointer to the function, the function being located outside the

access plan, which implements the operation indicated by the op. code, thus removing the interpretive step. This means that later on, when the access plan is being executed at run-time, instead of interpreting the op. code, the pointer is used to call the replacement code, thus a term "direct call", which is described in much more detail in the Specification.

Thus, the claims are directed to preprocessing, which happens prior to execution, which will allow a run-time improvement of execution of an existing and previously optimized access plan whose steps have already been determined and are not changed by the present invention but are substituted with pointers (Spec. p. 7, li. 5-10) in order to increase the run-time speed. As can be seen on p. 7, li. 25, the pre-processing is performed by the Access Plan Manager prior to storing the improved access path plan into a memory cache and before the execution. As can be seen on p.8, li. 14-16, this processed code section has pointers and is ready for execution, described on p. 8, li. 17-29.

Applicant respectfully objected to the very lengthy and numerous prior art paragraphs, sometimes over a column long, used to reject claims of the present invention because the Office Actions lack the required specificity of line number or element number and a corresponding element from the rejected claims of the present invention, thus making rejections overly broad and vague and, thus, weak. Therefore, responses to Office Actions respectfully requested that a more specific and exact references be used in the future because this kind of citation is not permitting a direct argument on point for each rejection, but to no avail. Moreover, there are many extraneous elements from the Chow reference in the Office Action that have nothing to do with the claimed invention which are cluttering the Office Action and showing lack of understanding of both the present invention and the reference. This was not corrected either, thus continuing with the total misrepresentation of the present invention and cited prior art.

It is true that Chow reference mentions access plan, interpreter, data structures, etc. These components are mentioned in most database applications. However, Chow reference is from a different field and performs a different function. It is directed to a method used to extend a SQL compiler for handling control statements packaged with query statements (Title, Specification). As Figs. 1-6 show, the end result of this reference is an access plan ("executable plan") and it teaches compilation of query statements (col. 1-8, col. 9 li. 1-21). Control statements are WHILE and REPEAT statements which are performed as a loop (col. 10, li. 12-18, col. 12, li 10-21, col.

17, li. 1-28). Sections in col. 14, li. 37-67 to col. 15, li. 1-67, col. 17 li. 29-67, col. 18, li. 13-39 describe a Pre-processing. However, in Chow reference it is pre-processing of a skeleton of control statements which is performed before compilation and before an access plan is created (specifically, in col. 18, li. 13-24). A lookup table in this reference is a symbol lookup table which stores variables and not op. code, as shown in numerous places such as, col. 18, li. 37-39, col. 20, li. 1-47, col. 24, li. 36-53. Sections of cols. 19-25 describe the work of a parser and compiler. The parser creates a syntax tree AST whose nodes are accessed by node pointers and stored in a stack. Col. 25, li. 1-8 describes that variables are indexed to maps and stored in data structures. Pointers to the variable data structure are stored in a symbol table entry. None of these are relevant to the present invention and thus Chow does not anticipate the independent claims 1, 10 and 19 of the present invention.

Regarding preamble, it is noted with appreciation that Office Action itself stated, on top of p. 4, that Chow is directed to a pre-processing of a query, by a compiler, for generating an executable plan. Moreover, it states that interpreter there is looking up variables. Therefore, Office Action itself held that Chow does not teach pre-processing of an access plan but of a query, before an access plan is even made, and a direct call is not even mentioned. Further, a Compiler, as used in Chow reference, creates an executable file by compiling a whole program with all code lines at once and later, during execution time, the executable file code lines get executed. However, present invention is directed to RDBMS with an interpreter and not a compiler; an Interpreter does not compile a whole program and does not create an executable file because it interprets and immediately executes code lines one at the time. Furthermore, element 118 of Chow is a threaded code generator and not an op. code generator and abstract, col. 1, col. 6. col. 8 and col. 10 citations are not relevant to the present invention. Therefore, the preamble of the independent claims 1, 10 and 19 of the present invention is also not anticipated by Chow.

Regarding the bottom of p. 4 of Office action, it states: "determining from the access plan an executable function associated with a first operation code (fig. 1, parser, the optimized QGM, and threaded code generation are used to produce operation code of SQL query statements to get the optimized access plan or access execution plan : figs. 1 and 2, item 115 & 118, col. 6, lines 25-67 and col. 8, lines 8-58, col. 14, lines 40-67 and col. 15, lines 1-67 and col. 16, lines 1-67);".

This language is a misrepresentation because it is taken from element (a) of independent claims of the present invention and it cannot be found in Chow reference. Chow is not determining anything from an access plan because its final product is an access plan, as shown above. As also shown above, cited sections are teaching use of a parser and compiler, before an access plan is created. Therefore, Chow does not teach an analysis of an existing access plan and determining from it an executable function associated with an op. code, as claimed in element (a) of the present invention.

Regarding the bottom of p. 4 of Office action, it states: "augmenting said first operation code in the access plan with a pointer to said executable function to provide a direct call mechanism replacing lookup function of a runtime interpreter (figs. 1 & 2, the extracted SQL statement or control statements is used to produce opcode based on parser, QGM and code generation and replacing this opcode (fig.2 and col. 15, lines 45-67 and col. 17, lines 25-67); the executable function in the run-time interpreter with a compiler generated local variables storing in symbol table for looking up with referencing local variables or pointers or indexes: fig. 4, Col. 24, lines 36-67 and col. 25, lines 1-8; also, col. 15, lines 1-67, col. 17, lines 1-28, Col. 20, lines 1-45 and col. 21, lines 5-55)."

This language is a misrepresentation because it is taken from element (b) of independent claims of the present invention and it cannot be found in Chow reference. As discussed above, Chow does not even mention augmenting or replacing op. codes in the access plan nor the direct call mechanism. It is noted with appreciation that Office Action on top of p. 5 held that Chow teaches that a look up is connected with a symbol table which references local variables. Therefore, Chow does not teach augmenting an op. code in the access plan with a pointer to an executable function, as claimed in element (b) of the present invention but a variable symbol table. Therefore, the elements (a) and (b) of the independent claims 1, 10 and 19 of the present invention are also not anticipated by Chow.

Moreover, Chow reference does not have any and all elements from the independent claims of the present invention. It does not pre-process an existing access plan, does not use a direct call mechanism, does not replace a lookup function of a run-time interpreter, does not include pointers to functions, does not use the normal executable function from an already generated

access plan (step (a)), does not create a new access plan with pointers to the same executable function (step (b)), etc.

Thus, the Chow reference clearly does not teach and is not directed to a pre-processing method replacing a lookup function of a run-time interpreter, which is claimed in the present invention, because these and other features of the present invention, recited in the claims and throughout the Specification, are not shown in this reference. Moreover, the cited reference does not show any features of the present invention, operating in the same way and for the same purpose. Thus, the reference is from a different art field.

Therefore, Office Actions have not provided the evidence which is needed to support rejections. Both the present invention and prior art were misinterpreted and the reasoning sweeps too wide and is without basis in law. Conclusionary statements of similarity without any articulated rationale or support do not constitute sufficient factual findings when not proven as similarity of structure, properties and utilities. Prior art relied upon does not render the present invention unpatentable because it fails to teach or suggest each and every element of the independent claims of the present invention and thus does not satisfy all limitations of the claimed present invention and is not pertinent to the present invention. Because Office Actions have not met the burden of proof for anticipation, since they did not establish prima facie case requested by 35 U.S.C. section 102(e), the application is patentable and entitled to grant of the patent.

ii) Prima Facie Case of Anticipation by Prior Art Has Not Been Established

To establish prima facie case of anticipation of a claimed invention, the standard for determining novelty under 35 USC 102 was been set forth by the courts *In re Bartleb*, 300 F.2d 942, 133 USPQ 204 (CCPA 1962), and *Hupp v. Siroflex of America Inc.*, 122 F.3d 1456, 43 USPQ2d 1887 (Fed. Cir. 1997), which states that the reference must be identical in all material aspects.

MPEP Sec. 2131, under the title TO ANTICIPATE A CLAIM, THE REFERENCE MUST TEACH EVERY ELEMENT OF THE CLAIM, states that "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete

detail as is contained in the ... claim." Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

With respect to claims 1, 10 and 19, Office Action stated: "Chow teaches a method for pre-processing an access plan generated for a query in a relational database management system to include said access plan including a plurality of operation codes, each of said operation codes being associated with one or more executable functions for performing the query (figs. 1-3, Pre-processing a SQL query statement is processed through a query compiler for generating a access executable plan for a database system such as IBM's DB2; this process requiring a run-time interpreter for looking up local variable; also this executable plan includes a plurality of SOD Query control statements, which are parsing into some operation codes into query graph model (QGM) representation of the statements, from which they are then processed to optimized QGM, an access execution plan such as cost-based optimized access plan and they are to be produced operation code or opcode by code generation module (item 118): abstract, col. 1, lines 15-35, col. 6, lines 25-67 and col. 8, lines 8-58; also col. 10, lines 1-40); determining from the access plan an executable function associated with a first operation code (fig. 1, parser, the optimized QGM, and threaded code generation are used to produce operation code of SQL query statements to get the optimized access plan or access execution plan : figs. 1 and 2, item 115 & 118, col. 6, lines 25-67 and col. 8, lines 8-58, col. 14, lines 40-67 and col. 15, lines 1-67 and col. 16, lines 1-67); and augmenting said first operation code in the access plan with a pointer to said executable function to provide a direct call mechanism replacing lookup function of a runtime interpreter (figs. 1 & 2, the extracted SQL statement or control statements is used to produce opcode based on parser, QGM and code generation and replacing this opcode (fig.2 and col. 15, lines 45-67 and col. 17, lines 25-67); the executable function in the run-time interpreter with a compiler generated local variables storing in symbol table for looking up with referencing local variables or pointers or indexes: fig. 4, Col. 24, lines 36-67 and col. 25, lines 1-8; also, col. 15, lines 1-67, col. 17, lines 1-28, Col. 20, lines 1-45 and col. 21, lines 5-55)."

As shown above, this highly incoherent statement is full of misinterpretations and completely fails to even mention the material aspect of the claims 1, 10 and 19: "direct call replacing a lookup function of the run time interpreter", from the preamble of these claims. As discussed above, Chow does not even mention augmenting or replacing op. codes in the access plan nor the direct call mechanism. Moreover, Chow reference does not have any and all elements from the

independent claims of the present invention. It does not pre-process an existing access plan, does not use a direct call mechanism, does not replace a lookup function of a run-time interpreter, does not include pointers to functions, does not use the normal executable function from an already generated access plan (step (a)), does not create a new access plan with pointers to the same executable function (step (b)), etc. Therefore, Office Actions do not follow the 102(e) rejection standard for determining novelty which states that the reference must be identical in all material aspects.

Thus, the Chow reference clearly does not teach and is not directed to a pre-processing method replacing a lookup function of a run-time interpreter, which is claimed in the present invention, because these and other features of the present invention, recited in the claims and throughout the Specification, are not shown in this reference. Moreover, the cited reference does not show any features of the present invention, operating in the same way and for the same purpose. Thus, the reference is from a different art field.

As shown above, Office Actions fail to show the identical invention in all material aspects in as complete detail in Chow reference as is contained in independent claims 1, 10 and 19. Therefore, it is clear that the standard for determining novelty under 35 USC 102(e) to establish prima facie case of anticipation has not been met for the claimed invention, because the cited reference does not have each and every element of independent claims but have none, is from a different field, has different components, works in a different mode of use and produces different results. Thus, independent claims 1, 10, and 19 and all claims dependent upon them in the present invention recite novel structure and therefore distinguish over the cited prior art, Chow, and are not anticipated by it under 35 U.S.C. 102(e).

<u>iii) The Examiner Uses Impermissible Hindsight to Modify the Teachings of Chow</u>

It is impermissible to use "hindsight reconstruction to pick and chose among isolated disclosures in the prior art to deprecate the claimed invention." *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

The Examiner did not specifically point out, as is required, why the rejected claims are deemed anticipated and where all of the specific limitations recited in the claims of the present invention

are found in the prior art. Examiner relies on Chow for disclosing all the limitations of the claims 1, 10 and 19 and their dependent claims but fails to point out how his statements, even if existent in the cited prior art, are related to the specific limitations of the claimed present invention. The Examiner's position is that it would be obvious and he merely asserts it without any explanation. However, the Chow reference fails to teach elements of the independent claims of the present invention, as shown above.

Moreover, in order to satisfy rejection of a limitation of a claim, the Examiner uses very broad and vague language without specifically pointing out any line where such information could be found in the long columns which we mostly cited whole. Further, Examiner uses statements unfound in the prior art and impermissible hindsight to modify the teachings. It is an impermissible hindsight by the Examiner that the teachings of Chow can be interpreted the way he presented them because nothing in the reference points to such interpretations. Moreover, the Examiner was impermissibly picking and choosing isolated disclosures of conventional database terms in the prior art to assert that the claims are unpatentable. Throughout all Office Actions the Examiner appears to argue that because a reference discloses the conventional terms well known and used in every write-up about database management systems that the reference anticipates the claims of the present invention.

Therefore, Chow reference cannot be used to invalidate independent claims 1, 10 and 19, and their dependent claims because it fails to teach each and every step of these claims. Thus, independent claims 1, 10 and 19 and all claims dependent upon them in the present invention recite novel structure and therefore distinguish over the cited prior art, Chow, and are not anticipated by it under 35 U.S.C. 102(e).

Sec. 102(e) Rejection of Claims 2 and 11

With respect to claims 2 and 11, Office Actions stated that Chow teaches comprising repeating steps (a) and (b) for remaining operation codes in the access plan (repeating the process with the SQL query statements with loop statement each time the function statement is called: Col. 10, lines 12-18 and col. 12, lines 10-22).

This is also a misinterpretation because Chow does not teach comprising repeating steps (a) and (b) for remaining operation codes in the access plan, because this language before the bracket is taken from the claimed present invention. As argued above, the cited sections are directed to the WHILE statement of a SQL query.

Therefore, it is clear that the standard for determining novelty under 35 USC 102 to establish prima facie case of anticipation has not been met for the claimed invention, because the cited reference does not have each and every element of independent claims and claims 2 and 11 but have none, is from a different field, has different components, works in a different mode of use and produces different results. Thus, dependent claims 2 and 11 of the present invention recite novel structure and therefore distinguish over the cited prior art, Chow, and are not anticipated by it under 35 U.S.C. 102(e).

Sec. 102(e) Rejection of Claims 3, 12 and 20

With respect to claims 3, 12 and 20, Office actions stated that Chow teaches wherein said step (b) comprises augmenting said first operation code in the access plan with a pointer to an intermediate function, said intermediate function including a data structure for storing a pointer to said executable function (replacing these opcode by a lookup function to look referencing variables storing in the symbol table as the value of addressing or pointers: fig. 4, col. 24, lines 36-67 and col. 25, lines 1-8).

This is also a misinterpretation because the language before the bracket is taken from the present invention. Moreover, Chow does not teach use of pointers to the op. code but to the variables, as held in the Office action.

Therefore, it is clear that the standard for determining novelty under 35 USC 102 to establish prima facie case of anticipation has not been met for the claimed invention, because the cited reference does not have each and every element of independent claims and claims 3, 12 and 20 but have none, is from a different field, has different components, works in a different mode of use and produces different results. Thus, dependent claims 3, 12, and 20 of the present invention recite novel structure and therefore distinguish over the cited prior art, Chow, and are not anticipated by it under 35 U.S.C. 102(e).

Sec. 102(e) Rejection of Claims 4, 13 and 21

With respect to claim 4, 13 and 21, Office Actions stated that Chow teaches wherein said data structure includes means for storing information associated with said executable function for said first operation code (fig. 4, symbol table, item 130 & 119, col. 24, lines 36-67).

This is also a misinterpretation because the language before the bracket is taken from the present invention. Moreover, Chow does not teach use of pointers to the op. code but to the variables, as held in the Office action.

Therefore, it is clear that the standard for determining novelty under 35 USC 102 to establish prima facie case of anticipation has not been met for the claimed invention, because the cited reference does not have each and every element of independent claims and claims 4, 13 and 21 but have none, is from a different field, has different components, works in a different mode of use and produces different results. Thus, dependent claims 4, 13, and 21 of the present invention recite novel structure and therefore distinguish over the cited prior art, Chow, and are not anticipated by it under 35 U.S.C. 102(e).

Sec. 102(e) Rejection of Claims 5, 14 and 22

With respect to claim 5, 14 and 22, Office Actions stated that Chow teaches wherein said step (b) comprises augmenting said first operation code in the access plan with a second pointer to a data structure, said data structure providing means for storing information associated with said first operation code or said executable function (in the symbol table has more than one local variables, each for each SQL query plan, thus there is a second pointer associated with the local variable or operation code: fig. 4, col. 24, lines 36-67).

This is also a misinterpretation because the language before the bracket is taken from the present invention. Moreover, Chow does not teach use of pointers to the op. code but to the variables, as held in the Office action.

Therefore, it is clear that the standard for determining novelty under 35 USC 102 to establish prima facie case of anticipation has not been met for the claimed invention, because the cited reference does not have each and every element of independent claims and claims 5, 14 and 22 but have none, is from a different field, has different components, works in a different mode of use and produces different results. Thus, dependent claims 5, 14, and 22 of the present invention recite novel structure and therefore distinguish over the cited prior art, Chow, and are not anticipated by it under 35 U.S.C. 102(e).

Sec. 102(e) Rejection of Claims 6 and 15

With respect to claim 6 and 15, Office actions stated that Chow teaches wherein said step (a) further includes assessing the executable function associated with the first operation code and if applicable, replacing the call to the executable function with a call to a second executable (call statement invoking the SQL statement and replacing the data statement: col. 14, lines 38-67 and col. 15, lines 1-67).

This is also a misinterpretation because the language before the bracket is taken from the present invention. Moreover, sections in col. 14, lines 38-67 and col. 15, lines 1-67 are directed to pre-processing of the control statements from a SQL query and not to the preprocessing of an existing access plan and its op. codes and functions.

Therefore, it is clear that the standard for determining novelty under 35 USC 102 to establish prima facie case of anticipation has not been met for the claimed invention, because the cited reference does not have each and every element of independent claims and claims 6 and 15 but have none, is from a different field, has different components, works in a different mode of use and produces different results. Thus, dependent claims 6 and 15 of the present invention recite novel structure and therefore distinguish over the cited prior art, Chow, and are not anticipated by it under 35 U.S.C. 102(e).

Sec. 102(e) Rejection of Claims 7 and 16

With respect to claims 7 and 16, Office Actions stated that Chow teaches wherein said intermediate function includes processing operations for the first operation code or the

executable function associated with the first operation code (col. 23, lines 45-67 and col. 24, lines 1-42).

This is also a misinterpretation because the language before the bracket is taken from the present invention. Moreover, sections in col. 23, lines 45-67 and col. 24, lines 1-42 are directed to a compilation and not to op. cods of an existing access plan.

Therefore, it is clear that the standard for determining novelty under 35 USC 102 to establish prima facie case of anticipation has not been met for the claimed invention, because the cited reference does not have each and every element of independent claims and claims 7 and 16 but have none, is from a different field, has different components, works in a different mode of use and produces different results. Thus, dependent claims 7 and 16 of the present invention recite novel structure and therefore distinguish over the cited prior art, Chow, and are not anticipated by it under 35 U.S.C. 102(e).

Sec. 102(e) Rejection of Claims 8 and 17

With respect to claims 8 and 17, Office Action stated that Chow teaches wherein said processing operations in the intermediate function include gathering statistics on the user of the executable function associated with the operation code (fig. 1, item 301, analyzer generating information for the control flow and scopes and symbol table such as statistical information: col. 18, lines 12-40).

This is also a misinterpretation because the language before the bracket is taken from the present invention. Moreover, the cited sections and Fig. 1 are not directed to op. codes of an existing access plan, symbol tables defines variables and not op. codes, and the claims are not directed to "control flow and scopes".

Therefore, it is clear that the standard for determining novelty under 35 USC 102 to establish prima facie case of anticipation has not been met for the claimed invention, because the cited reference does not have each and every element of independent claims and claims 8 and 17 but have none, is from a different field, has different components, works in a different mode of use and produces different results. Thus, dependent claims 8 and 17 of the present invention recite

novel structure and therefore distinguish over the cited prior art, Chow, and are not anticipated by it under 35 U.S.C. 102(e).

Therefore, claims 1-8, 10-17 and 19-22 are not anticipated by Chow reference under 35 U.S.C. 102(e).

(B) Sec. 103(a) Rejection of Claims 9 and 18

Claims 9 and 18 stand rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No.: 5,875,334 issued to Chow et al. (hereinafter Chow) in view of US Patent No.: 6,077,312 issued to Bates et al. ("Bates").

i) Prima Facie Case of Obviousness by Prior Art Has Not Been Established and the Chow and Bates References Would Not Satisfy All the Limitations of the Claims

To establish prima facie obviousness of a claimed invention, all the claims limitations must be taught or suggested by the prior art, MPEP Sec. 2143.03, citing *In re Royka*, 180 USPQ 580 (CCPA 1974).

As stated in MPEP Sec. 706.02(j), 35 U.S.C. 103 authorizes a rejection where, to meet the claim, it is necessary to modify a single reference or to combine it with one or more other references. After indicating that the rejection is under 35 U.S.C. 103, the examiner should set forth in the Office action:

(A) the relevant teachings of the prior art relied upon, preferably with reference to the relevant column or page number(s) and line number(s) where appropriate,
(B) the difference or differences in the claim over the applied reference(s),
(C) the proposed modification of the applied reference(s) necessary to arrive at the claimed subject matter, and
(D) an explanation why one of ordinary skill in the art at the time the invention was made would have been motivated to make the proposed modification.

To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See MPEP Sec. 2143 - 2143.03 for decisions pertinent to each of these criteria.

The initial burden is on the examiner to provide some suggestion of the desirability of doing what the inventor has done. "To support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed invention or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references." Ex parte Clapp, 227 USPQ 972, 973 (Bd. Pat. App. & Inter. 1985). See MPEP Sec. 2144 - 2144.09 for examples of reasoning supporting obviousness rejections.

With respect to claims 9 and 18, Office Actions stated that Chow teaches a method as discussed in claim 1, and that Chow teaches a SQL query statement is processed through a query compiler for generating a access executable plan for a database system such as IBM's DB2; this process requiring a run-time interpreter and for looking up local variable; also this executable plan includes a plurality of SQL3 Query control statement for parsing into some operation codes from query graph model (QGM) representation of the statement to produce an operation code based on access executable plan, replacing the executable function in the run-time interpreter with a compiler generated local variable for looking up the symbol table with referencing local variables or pointers.

As shown above, it is noted with appreciation that Office Actions themselves stated, on top of p. 4, that Chow is directed to pre-processing of a query, by a compiler, for generating an executable plan. Moreover, it states that interpreter there is looking up variables. Therefore, Office Actions held that Chow does not teach pre-processing of an access plan but of a query, before an access plan is even made, and a direct call is not even mentioned. Therefore, as proven above, the independent claims 1, 10 and 19 of the present invention are not anticipated by Chow.

Moreover, the Office Actions are not following the law. As shown by the Applicant, the teachings of the referenced prior art are not relevant to the claimed invention. As shown in Amendments, the proposed modifications of the applied reference(s) necessary to arrive at the claimed subject matter are not shown, and an explanation about why one of ordinary skill in the art at the time the invention was made would have been motivated to make the proposed modification was not given. Although Applicant argued in the Responses to Office Actions that a modification must be shown in the prior art itself, each Examiner's Office Action failed to address these points and follow the law. Regarding claims 1, 10 and 19, the Chow reference does not teach, show or suggest claimed subject of the present invention. Therefore, this reference cannot be used to invalidate independent claims 1, 10, and 19 and their dependent claims. Moreover, the Examiner quoted parts of sentences nonexistent in those references. However, even if these quotes are correct, the element must be pointed to in the prior art itself and no such

element or its modification is pointed to in the cited reference nor it could be since it performs differently. Therefore, Chow reference cannot be used to invalidate independent claims 1, 10 and 19 and their dependent claims because it fails to teach or suggest any and all the steps of these claims.

It is noted with appreciation that Office Action held that Chow does not clearly teach a pause for receiving user input before or after the call to the executable function.

Office Action, however, stated that Bates teaches halt execution of a computer program when the computer meets a predetermined criteria (abstract) and that, therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Chow the teachings of Bates, and that one having ordinary skill in the art would have found it motivated to utilize the halt execution of a program when it is needed as disclosed (Bates' abstract), into the system of Chow for the purpose of including a pause for executing a program to replace opcode, thereby, helping to locate and identify errors in a program under development (Bates' col. 1, lines 10-20).

Claims 9 and 18 are directed to inserting a deliberate pause for receiving user input before or after the call to the executable function. It is accepted with gratitude that Office Action itself held that Bates teaches to halt execution of a computer program when the computer meets a predetermined criteria. Thus, Office Action itself held that in Bates it is the decision of the computer program when to stop, i.e., when the computer meets a predetermined criteria, and not of the user. It also held that it halts (permanently stops) and does not just make a pause in order to receive the user's input. Moreover, Bates reference is directed to debugging utilizing a context sensitive breakpoints, which is a completely different field. Further, Bates is motivated to halt the execution of a program when it is needed to locate and identify errors in a program under development (Bates' col. 1, lines 10-20). However, the present invention is not directed to debugging so the motivation given in the Office Action to combine these references is clearly wrong.

Moreover, "would have been obvious" allegations used to reject claims of the present invention are only the unsupported conclusions not sufficient under Sec. 103 and Examiner failed to cite prior art references which support the "would have been obvious" allegations and show how modifications can be accomplished and what motivation was used to modify a reference to arrive

at the claimed subject matter and to show how this modified reference functions and which structure it has. Applicant challenged the Office Action practice, used to reject claims 9 and 18 of the present invention, under Sec. 103 and, as allowed under MPEP Sec. 2144.03, respectfully requests that Examiner cites prior art references which support all these "would have been obvious" allegations and show how modifications can be accomplished in the cited reference and what motivation was used to modify a reference to arrive at the claimed subject matter and to show how this combination of modified references functions and which structure it has and how it makes the claimed present invention obvious, but to no avail.

Moreover, Applicant includes herein the same argument as regarding claims 1, 10 and 19 because Bates reference shows a different system and method from the present invention and so does the Chow reference, as shown above, and neither reference is directed to the direct call threaded code of the present invention. Therefore, the references cannot be combined and be used to reject claims 9 and 18, which recite novel and unobvious structure and use and, therefore, distinguish over the cited prior art, Chow and Bates.

Further, the cited references Chow and Bates are individually complete and they do not suggest a modification. Case *Amgen, Inc. v. Chugai Pharmaceutical Co.*, 927 F.2d 1200, 18 USPQ2d 1016 Fed. Cir. 1991) is on point as is the case *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984), which held that where there is no technological motivation for a modification or if a proposed modification of reference would destroys intent, purpose or function of the reference, the prima facie case of obviousness is not properly established. This law was not followed in the Office Actions either.

Further, the Examiner has not established a prima facie case of obviousness because the three basic criteria, which must be met, were not met. Office Action has no suggestion or motivation, either in the reference or in the knowledge generally available to one of ordinary skill in the art to modify the reference or to combine reference teachings, a reasonable expectation of success was not shown (and is impossible) and the prior art reference(s), which must teach or suggest all the claim limitations, does not do so here. Furthermore, the Examiner did not satisfy the initial burden to provide some suggestion in the references of the desirability of doing what the inventor has done, because to support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed

invention or the Examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references. Furthermore, the cited reference is from nonanalogous art, under MPEP Sec. 2141.01(a).

These references are from completely different fields unrelated to a pre-processing of access plans using pointers and the methods taught by these two references are among themselves from completely different art fields and cannot be combined. Moreover, it is shown that they do not perform any elements of the independent claims 1, 10, and 19 and therefore their dependent claims. Further, they satisfy a different need from a different area and do not teach pre-processing of existing access plan. Therefore, these references cannot be used to invalidate independent claims 1, 10, and 19 and their dependent claims. Because none of the referenced prior art teaches elements (a) to (b) of claims 1, 10 and 19, which are the main steps of the present invention, their combination is not a valid reason for rejection of these independent claims and claims dependent thereof. Therefore, each cited reference, by itself or in combination, cannot be used to invalidate claims 1, 10 and 19 because they fail to teach any and all the steps of these claims.

Therefore, Chow and Bates references cannot be used to invalidate independent claims 1, 10 and 19, and their dependent claims 9 and 18 because they fail to teach any and all the steps of these claims. Thus, Office Actions themselves show that the claims of the present invention are unobvious and are, thus, patentable.

8. Evidence Appendix

No such evidence exists.

9. Related Proceedings Appendix

There are no related proceedings.

10. Conclusion

Regarding claims 1-22, none of the cited references teaches, shows, suggests or is even remotely related to pre-processing of existing access plans with a direct call mechanism to augment op. code with pointers, as claimed by the present invention. Therefore, these reference cannot be used to invalidate independent claims 1, 10, and 19 and their dependent claims. Moreover, the Examiner combined references from different arts in order to reject claims 1-22, by quoting parts of sentences nonexistent in those references. However, even if these quotes are correct, the combination must be pointed to in the prior art itself and no such combination is pointed to in the cited references nor it could be since they are from different fields. Therefore, these references cannot be used to invalidate independent claims 1, 10 and 19 and their dependent claims because they fail to teach any and all the steps of these claims.

Improper interpretation and/or combination of cited references is used in each claim rejection in the Office Action. None of the cited references suggests combination under In re Sernaker, 217 U.S.P.Q. 1, 6 (CAFC 1983), and one skilled in the art would have no reason to make a combination since they are from different fields, impossible to combine and individually complete. Moreover, none of the cited references discloses the subject matter and features of claims 1-22 of the present invention and even if they did show some individual features, they would not be able to meet the claims of the present invention which provide new and unexpected results over these references and are thus novel and unobvious and patentable under Sections 102 and 103.

Thus, it is respectfully submitted that the cited prior art does not render claims 1-22 unpatentable. In view of the foregoing, it is submitted that the final rejections of claims 1-22 are improper and, accordingly, the Board is respectfully requested to reverse the final rejections and order that this application proceed immediately to issue.

Respectfully submitted,

Date: December 30, 2006

Sandra M. Parker
Reg. No. 36,233

LAW OFFICE OF SANDRA M. PARKER
329 La Jolla Avenue
Long Beach, CA 90803
Phone/fax: (562) 597-7504

CLAIMS APPENDIX

1. **(Previously Amended)** A method for pre-processing an access plan generated for a query in a relational database management system to include a direct call mechanism replacing a lookup function of a run-time interpreter, said access plan including a plurality of operation codes, each of said operation codes being associated with one or more executable functions for performing the query, said method comprising the steps of:

(a) determining from the access plan an executable function associated with a first operation code; and

(b) augmenting said first operation code in the access plan with a pointer to said executable function to provide a direct call mechanism replacing a lookup function of a run-time interpreter.

2. **(Original)** The method as claimed in claim 1, further comprising repeating steps (a) and (b) for the remaining operation codes in the access plan.

3. **(Original)** The method as claimed in claim 1, wherein said step (b) comprises augmenting said first operation code in the access plan with a pointer to an intermediate function, said intermediate function including a data structure for storing a pointer to said executable function.

4. **(Original)** The method as claimed in claim 3, wherein said data structure includes means for storing information associated with said executable function or said first operation code.

5. **(Original)** The method as claimed in claim 1, wherein said step (b) comprises augmenting said first operation code in the access plan with a second pointer to a data structure, said data structure providing means for storing information associated with said first operation code or said executable function.

6. **(Original)** The method as claimed in claim 1, wherein said step (a) further includes assessing the executable function associated with the first operation code and if applicable, replacing the call to the executable function with a call to a second executable function.

7. **(Original)** The method as claimed in claim 3, wherein said intermediate function includes processing operations for the first operation code or the executable function associated with the first operation code.

8. **(Original)** The method as claimed in claim 7, wherein said processing operations in the intermediate function include gathering statistics on the use of the executable function associated with the operation code.

9. **(Original)** The method as claimed in claim 7, wherein said processing operations in the intermediate function include a pause for receiving user input before or after the call to the executable function.

10. **(Previously Amended)** A computer program product for use on a computer wherein queries are entered by a user for retrieving data in a relational database management system having a query optimizer for generating an access plan for executing the query, said query optimizer including a direct call mechanism replacing the lookup function of a run-time interpreter, said computer program product comprising:

a recording medium;

means recorded on said recording medium for instructing said computer to perform the steps of:

(a) determining an executable function associated with a first operation code in the access plan, the first operation code being one of a plurality of operation codes; and

(b) augmenting said first operation code in the access plan with a pointer to said executable function to provide a direct call mechanism replacing a lookup function of a run-time interpreter.

11. **(Original)** The computer program product as claimed in claim 10, the means for instructing said computer further comprising repeating steps (a) and (b) for the remaining operation codes in the access plan.

12. **(Original)** The computer program product as claimed in claim 10, wherein said step (b) comprises augmenting said first operation code in the access plan with a pointer to an intermediate function, said intermediate function including a data structure for storing a pointer to said executable function.

13. **(Original)** The computer program product as claimed in claim 12, wherein said data structure includes means for storing information associated with said executable function or said first operation code.

14. **(Original)** The computer program product as claimed in claim 10, wherein said step (b) comprises augmenting said first operation code in the access plan with another pointer to a data structure, said data structure providing means for storing information associated with said first operation code or said executable function.

15. **(Original)** The computer program product as claimed in claim 10, wherein said step (a)

further includes assessing the executable function associated with the first operation code and if applicable, replacing a call to the executable function with a call to another executable function.

16. **(Original)** The computer program product as claimed in claim 12, wherein said intermediate function includes processing operations for the first operation code or the executable function associated with the first operation code.

17. **(Original)** The computer program product as claimed in claim 16, wherein said processing operations in the intermediate function include gathering statistics on the use of the executable function associated with the first operation code.

18. **(Original)** The computer program product as claimed in claim 12, wherein said processing operations in the intermediate function include a pause for receiving user input before or after a call to the executable function.

19. **(Previously Amended)** A relational database management system for use with a computer system wherein queries are entered by a user for retrieving data from tables, the relational database management system including a query optimizer for generating an access plan associated with the queries entered by the user, said query optimizer including a direct call mechanism replacing a lookup function of a run-time interpreter, said relational database management system comprising:

(a) means for determining an executable function associated with each of a plurality of operation codes in the access plan; and

(b) means for augmenting said operation codes in the access plan with a pointer to

said executable function associated with each operation code to provide a direct call mechanism replacing a lookup function of a run-time interpreter.

20. **(Original)** The relational database management system as claimed in claim 19, wherein said means for augmenting said operation codes includes means for replacing said operation codes in the access plan with a pointer to an intermediate function, said intermediate function including a data structure for storing a pointer to said executable function.

21. **(Original)** The relational database management system as claimed in claim 20, wherein said data structure includes means for storing information associated with said executable function or said operation codes.

22. **(Original)** The relational database management system as claimed in claim 19, wherein said means for augmenting said operation codes includes means for adding another pointer to a data structure, said data structure providing means for storing information associated with said operation codes or said executable function.

EVIDENCE APPENDIX

NONE

RELATED PROCEEDINGS APPENDIX

NONE